Implementation of Trust Rank Algorithm

Aayush Shrivastava
CS24MTECH02001

Anish Pawar
AI23MTECH14002

Ankit Pandey
AI23MTECH13002

Puneet Kumar Rajan
AI23MTECH11010

Vivek Tarachandani
AI23MTECH14008

This report is included in the coursework for Assignment 2 of the Fraud Analytics (CS6890) course.

Code: https://colab.research.google.com/drive/1Tx3D9y98Pp8ratEBA-WfwDe-MDYvY4df?usp=sharing

1. Problem Statement

The aim is to pinpoint fraudulent nodes within a network characterized by edge weights representing transaction strength among nodes. This entails devising a method capable of effectively detecting nodes exhibiting malicious or undesirable behavior, relying on discernible patterns within node connections. The devised approach must be scalable for large-scale networks, necessitating implementation within a distributed computing framework. This project employs Trust Rank to identify problematic nodes, initializing the process with an Oracle's predefined set of suspect nodes. Consequently, nodes exhibiting unfavorable behavior will receive higher ranks, while those deemed trustworthy will rank lower. Conceptually akin to distrust rank, this methodology is implemented utilizing the `Pregel` framework to address scalability concerns.

2. Dataset Description

The provided dataset contains two CSV files, namely `'Payments.csv'` and `'bad_sender.csv'`. The former contains the data of transactions between two parties with the columns titles 'Sender,' 'Receiver', and 'Amount.' The latter contains the IDs of all the bad nodes given to us by an oracle. Initially, the IDs in both datasets were in the range [1001, 2190]. For the sake of convenience, they were scaled between 0-800, which is the number of unique IDs in both datasets.

Some of the nodes contained multiple edges, to consolidate them, we added the amount sent between them. In case of leakage (nodes with no out link), we created synthetic links to bad nodes with equal weightage.

- `Payments.csv`: Contains the transaction amount between Sender and Receiver.
  - **Sender**: Node ID of the Sender
  - **Receiver**: Node ID of the Receiver
  - **Value**: Contains the transaction amount
- `bad_sender.csv`: Contains a list of bad nodes given by Oracle.
  - **'Bad Sender'**: Node IDs of bad senders

3. Algorithm

3.1. TrustRank

TrustRank is an algorithm that enhances search engine result quality by combating web spam. Initially, a select group of reputable "seed" pages is manually identified. These pages are assigned high trust scores, which are then propagated iteratively through the web's link structure. Pages linking to trusted ones inherit higher trust scores, contributing to a recursive spread of trust. TrustRank calculates the trustworthiness of each page based on these scores, with a threshold used to differentiate trustworthy from untrustworthy pages. Pages surpassing this threshold are prioritized in search engine rankings, thereby promoting high-quality content and filtering out spam.

3.2. NetworkX

In our TrustRank implementation, NetworkX serves as a fundamental tool for constructing and analyzing complex networks. Leveraging its versatile functionalities, we efficiently represent the network of nodes and their interconnections, which are crucial for TrustRank computation. NetworkX provides intuitive methods for graph manipulation, facilitating the identification of patterns in node connections essential for distinguishing trustworthy and sus-

Algorithm 1 TrustRank Algorithm

Require: N: Total number of web pages
Require: M: Set of pages that have outgoing links
Require: L_i: Number of outgoing links on page i
Require: T_i: Trust score of page i
Require: α: Damping factor (typically $\alpha \in (0.8, 0.9)$)

procedure TRUSTPROPAGATION
  Identify a small set of seed pages as trusted.
  Initialize trust scores T_i to 1 for seed pages.
  for each page i with outgoing links **do**
    $T_i^{(t+1)} \leftarrow \alpha \cdot \sum_{j \in M_i} \frac{T_j^{(t)}}{L_j} + (1 - \alpha) \cdot \frac{1}{N}$ ▷ where M_i is the set of pages that link to page i
  end for
end procedure

procedure FINALSCORECALCULATION
  Combine trust and untrust scores to determine the final score for each page.
  Typically, the final score is calculated as $T_i - U_i$ or using some other method that incorporates both trust and untrust values.
end procedure

picious nodes. It provides robust algorithms that streamline the calculation of node rankings based on TrustRank principles, effectively identifying bad nodes within the network. Additionally, NetworkX seamlessly integrates with other Python libraries, enhancing our ability to develop a scalable and adaptable TrustRank solution.

3.3. Training

The algorithm was trained for 100 iterations with a damping factor = 0.85.

4. Results and Discussion

The algorithm takes a directed graph as the input, computes the 'trust score' for each node, and gives it as the output.

Since the ground truth isn't available in the dataset, we have plotted the 'bad nodes,' provided by Oracle, to observe a trend in the algorithm's output by visual inspection.

From Figure 1, it's evident that the bad nodes have a higher score, indicating that our algorithm is working as intended. A heuristic threshold of 0.01 was set to segregate the bad nodes from the rest.

From Figure 2, it's evident that the majority of the nodes have a score below 0.01, which is the heuristic threshold set by us. We can conclude that the data contains very few bad nodes.

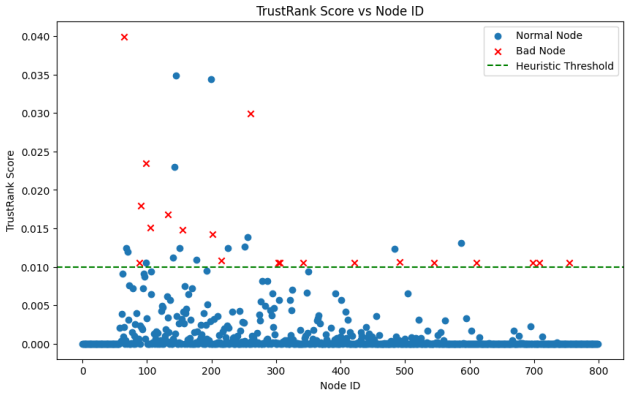

Figure 1. TrustRank Score vs Node ID (Bad nodes are highlighted separately)

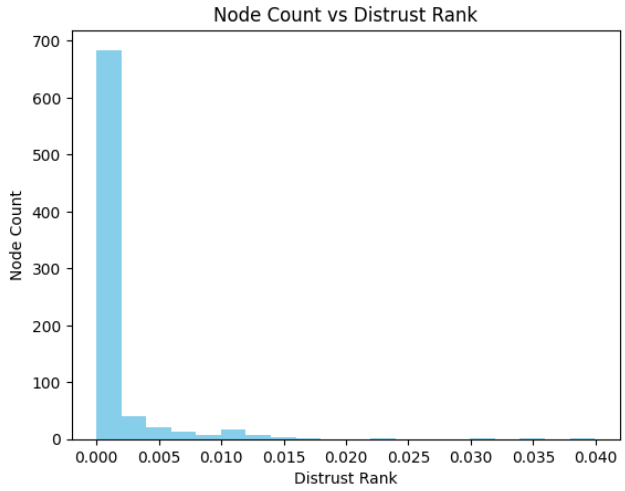

Figure 2. Frequency Plot of Node Count vs Distrust Rank

5. Conclusion

The implementation of the TrustRank algorithm using NetworkX has proven to be an effective approach in combating web spam and improving the quality of search engine results. By leveraging the web's link structure and propagating trust scores through the network, we identified trustworthy pages and prioritized them in search engine rankings. Through the iterative process of trust propagation and the incorporation of seed pages, our implementation successfully distinguished between reputable and untrustworthy sources. This not only enhances the relevance and reliability of search results but also contributes to a more positive user experience by delivering high-quality content. Moving forward, further refinements and optimizations to the TrustRank algorithm can continue to strengthen its effectiveness in combating web spam and promoting trustworthiness on the web.